

## **REMARKS/ARGUMENTS**

The Office Action dated March 5, 2008 has been reviewed and carefully considered.

Reconsideration of the above-identified application, as herein amended, is respectfully requested.

### **Status of the Application**

Pending claims 1 to 9, claims 1 and 8 being independent, remain pending in this application. By this Amendment, claims 1, 7, 8, and 9 have been amended. The amendments made to the claims do not alter the scope of the claims nor have these amendments been made to define over the prior art. Rather, the amendments to the claims have been made for cosmetic reasons to improve the form thereof. No new matter has been added.

In the Office Action of March 5, 2008, the Examiner rejected claim 7 under 35 U.S.C. §101 as being directed to non-statutory subject matter; and claims 1-9 under 35 U.S.C. §102(b) as allegedly anticipated by “Clustering Intrusion Detection Alarms to Support Root Cause Analysis” (“Julisch”). Applicants have carefully considered the Examiner’s rejections, together with the comments provided in support thereof, and respectfully traverse the rejections and submit that the invention as claimed is patentably distinct from the applied reference.

### **Rejection Under 35 USC §101**

The Examiner rejected claim 7 under 35 USC §101 as being directed to non-statutory subject matter. Applicants have amended claim 7 to explicitly recite “a computer readable medium encoded with a computer program.” Thus, claim 7 is directed towards statutory subject matter and this rejection should be withdrawn.

## **Prior Art Rejection**

The Examiner rejected claims 1-9 under 35 USC §102(b) as anticipated by Julisch.

Applicants respectfully request reconsideration and withdrawal of this rejection.

## **The Present Disclosure**

One aspect of the present invention relates to a simple method of unsupervised classification of alerts issued by intrusion detection sensors and generation of general alerts providing an overview of the alerts.<sup>1</sup> The disclosed method (and associated system) automatically classifies alerts issued by the intrusion detection sensor of a security system.

Each alert is defined by a plurality of qualitative attributes organized in a hierarchical structure and belonging to a plurality of attribute domains having a partial order relationship. A trellis based on attributes specific to each alert is constructed for each alert issued by the intrusion detection sensors. The specific trellis includes nodes corresponding to alerts that are linked to each other by arcs so that each node is linked to one or more parent nodes and/or one or more child descendent nodes. Each specific trellis is iteratively merged into a general trellis. Selecting the alerts that are simultaneously the most pertinent and the most general in accordance with statistical criteria and according to their attributes identifies the collated alerts in the general trellis. The collated alerts are output to provide an overview of the alerts issued by the intrusion detection sensors.

Applicants point out that a partial order relationship is used for each alert to construct a trellis specific to that alert by generalizing each alert according to each of its attributes and to all

---

<sup>1</sup> These descriptive details are provided only for the convenience of the Examiner as part of the discussion presented herein, and are not intended to argue limitations that are not claimed. Further, this is not intended to argue any interpretation of any claim term that is narrower than would be understood by one of ordinary skill in the art in the context of the specification and the claims as a whole.

levels of the hierarchical structure. (See specification as filed at page 11, lines 9-14.) A general trellis is constructed by successively adding specific trellises. An individual specific trellis is inserted into the general trellis by merging the specific individual trellis with the general trellis. Thus, a specific trellis is constructed for each alert issued by the intrusion detection sensor. Each specific trellis is then added to create the general trellis that contains all of the alert concepts. (See page 11, lines 20-25.)

For any generalizable attribute of a given alert, the generalized value of that attribute is recovered from its hierarchical structure to form a new alert more general than the given alert. For example, a new node corresponding to the new alert is formed in accordance with the generalization of a second attribute and the node is added to the specific trellis together with an arc extending from the new node of the new alert to the node of the given alert. Arcs between the parent node and the alert node are added.

The merging of a given specific trellis into a general trellis is shown in Figure 3 and discussed in the specification at page 14, line 26 *et seq.* In this example, parameters are defined for the general trellis. Specifically, a first node is selected corresponding to a first alert or concept belonging to the specific trellis and a second node is selected corresponding to a second alert or concept belonging to the general trellis. Next, an offspring node of the first node is selected. The system and method then verifies that the offspring node of the first node belongs to the general trellis. If the offspring node of the first node does belong to the general trellis, the arcs coming from parent nodes of the offspring nodes are eliminated; however, the nodes remain. The system algorithm is then recursively executed for the offspring with new parameters. If the offspring is not in the general trellis then it is added with its descendants before another offspring of the first node is selected.

In this manner, the general trellis is constructed by iteratively merging each specific trellis into the general trellis. This process retains the attributes from each specific trellis. In other words, specific alerts remain in both the specific trellis and the general trellis after generalization because alerts are not removed from the generalization algorithm after generalization.

#### Clustering Intrusion Detection Alarms to Support Route Cause Analysis

Julisch, cited by the Examiner, is a minor change to and rehashing of the 2001 article “Mining Alarm Clusters To Improve Alarm Handling Efficiency” which is discussed in the background art section of the present specification at page 3, line 12 *et seq.*

The system in Julisch modifies a preexisting iterative process that chooses an attribute and generalizes the attribute of each individual as a function of the associated hierarchy. Variables that are equal after generalization are merged. Thus, the overall number of variables decreases with every iteration. The iteration process stops when the number of variables falls below a given threshold. The generalized alerts may be over-generalized and of limited interest. Julisch does not generalize alerts for which the number of underlying alert instances is greater than a preset threshold. To avoid over-generalization, once the threshold is met, generalization of the remaining alerts is cancelled and the iteration process restarts using another attribute.

As presented in Julisch, an intrusion detection system uses a semiautomatic approach to generalize alerts from intrusion detection sensors. (See Julisch page 444, paragraph 3 to page 445, paragraph 1.) Causes of the alert are identified and a generalized alert to an operator is produced. The generalized alert uses an alert clustering method that groups similar alerts. Specifically, Julisch uses unsupervised classification of a series of alerts where each alert is defined by a plurality of attributes belonging to a plurality of attribute domains

In Julisch, a graph for each alert is constructed by generalizing the alert in accordance with its attributes. (See page 449.) Each graph has nodes that correspond to alerts that are linked to each other by arcs. Generalizing an alert occurs by selecting an attribute and generalizing this attribute by replacing it with a parent. (See page 456, paragraph 6.) This generalization is done iteratively until a predetermined threshold of generalization is reached. (See *id.*) The iteration process continues until an alarm has been found to which the least mean\_size of the original alarm can be generalized. Thus, an alarm is removed from the generalization algorithm once the alarm is generalized.

Julisch is unable to identify pertinent generalizations that might have arisen if the alert supplied to the security operator had been retained for subsequent generalizations. Moreover, the nature of the generalized alerts obtained depends on the order in which the attributes are considered, which is based on heuristics. Finally, the Julisch method is not incremental and the generalization process must be reinitialized on each request from the security operator. Thus, the specific alert conditions are not present in the generalization of Julisch.

Claims 1-9 are not Anticipated by Julisch

Independent claims 1 and 8 explicitly recite “iteratively merging each specific trellis into a general trellis.” As discussed above, Julisch uses a semiautomatic approach to generalized alerts, where each alert is defined by a plurality of attributes belonging to a plurality of attribute domains, from intrusion detection sensors. The generalized alerts are clustered using an alert clustering method that groups similar alerts. Julisch fails to iteratively merge each specific trellis into a general trellis.

While Julisch discloses a generalization of attributes, the generalization process stops when the number of instances of generalization exceeds a given threshold. In this manner, as

discussed in applicants' Background of the Invention, the alarm is either over-generalized or not generalized enough. In contrast, applicants' iterative merging of each specific trellis into the general trellis alerts without stop criteria is based on statistical criteria which allows for automatic generalization and results in more accurate generalization of alerts.

Another advantageous result of iteratively merging each specific trellis into the general trellis is that applicants' incremental process can identify relevant generalizations conserving generalized alerts provided to the operator. In contrast, in Julisch once the alarm is generalized it is removed from the generalization algorithm and the algorithm is reset for each attribute of a new alert. Thus, applicants' iterative merging of each specific trellis into the general trellis provides for the identification of generalizations because alarms are not removed from the generalization algorithm.

For those reasons, the cited Julisch reference fails to render claims 1 and 8 unpatentable.

Claims 2-7 depend from and contain all of the limitations of independent claim 1, and claim 9 depends from and contains all of the limitations of independent claim 8. These dependent claims also recite additional limitations which, in combination with the limitations of the independent claims from which they depend, are neither disclosed nor suggested by the cited reference and are also directed to patentable subject matter. Thus, claims 2-7 and 9 should also be allowed.

## **Conclusion**

In view of the foregoing, it is respectfully submitted that the all of the pending claims are patentably distinct over Julisch.

Applicants have responded to all of the objections and rejections cited in the Office Action. Reconsideration and a Notice of Allowance for all of the pending claims are therefore requested.

If the Examiner believes that an interview would be of assistance, the Examiner is encouraged to contact the undersigned at the number listed below.

It is believed that no additional fees or charges are required at this time in connection with the present application. However, if any such fees or charges are required at this time, they may be charged to our Patent and Trademark Office Deposit Account No. 03-2412.

Respectfully submitted,

COHEN PONTANI LIEBERMAN & PAVANE LLP

By \_\_\_\_\_

Lance J. Lieberman

Reg. No. 28,437

551 Fifth Avenue, Suite 1210

New York, New York 10176

(212) 687-2770

Dated: August 5, 2008